UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/535,765	03/28/2000	Vincent E. Hummel		5136

7590 06/12/2003

Blakely Sokoloff Taylor & Zafman LLP
12400 Wilshire Boulevard
7th Floor
Los Angeles, CA 90025

EXAMINER	
HUISMAN, DAVID J	
ART UNIT	PAPER NUMBER
2183	5

DATE MAILED: 06/12/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Applicati n No.** 09/535,765	**Applicant(s)** HUMMEL, VINCENT E.
	Examiner David J. Huisman	**Art Unit** 2183

-- The MAILING DATE of this communication appears on the cover she t with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 April 2003* .

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-9,11-16,18 and 19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-9,11-16,18 and 19* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 March 2000* is/are: a)□ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)□ The proposed drawing correction filed on _____ is: a)□ approved b)□ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)□ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____ .

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) □ The translation of the foreign language provisional application has been received.

15)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .

4) □ Interview Summary (PTO-413) Paper No(s). _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: .

DETAILED ACTION

1. Claims 1, 2, 4-9, 11-16, 18, and 19 have been examined.

Claim Rejections - 35 USC § 102

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1, 2, 6-8, and 14-15 are rejected under 35 U.S.C. 102(e) as being anticipated by McFarling, U.S. Patent No. 6,374,349 (as applied in the previous Office Action).

3. Referring to claim 1, McFarling has taught a method comprising:

a) providing a first taken/not-taken prediction responsive to an address using a saturating counter branch predictor. See column 9, lines 10-16. It has been disclosed that a bimodal predictor always provides a prediction. Furthermore, from column 3, lines 11-28, the bimodal predictor is shown to be an array of saturated counters.

b) providing (1) a second taken/not-taken prediction responsive to the address resulting in a hit in a local branch history table, and (2) a hit/miss indication for the address. See column 9, lines 18-24, and Fig.10. Note in Fig.10 that the hit indication controls which prediction is propagated through the multiplexer.

c) selecting for the address one of (1) the second prediction if the indication is a hit, and (2) the first prediction if the indication is a miss. Again, see column 9, lines 10-24, and Fig.10. If a history-table miss occurs, the counter prediction will be used. On the other hand, if a history-table hit occurs, the history table prediction will be used.

d) updating a replacement field for a matching entry in the local branch history table only if the first prediction is incorrect, indicating that the entry is used to make a prediction. See column 9, lines 45-52. When the first prediction from the bimodal predictor (saturating counter predictor) is correct, the corresponding entry in the history table is not updated. Therefore, replacements would only occur when the first prediction is incorrect.

4. Referring to claim 2, McFarling has taught a method as described in claim 1. McFarling has further taught hashing the address prior to indexing at least one of the saturating counter branch predictor and the local branch history table. See column 3, lines 12-16, and column 4, lines 15-19. Recall that hashing is simply the function of converting a value (in this case, the branch instruction address) into a value used for locating corresponding data in a structure (in this case, the saturating counter predictor and the local history predictor). In the system of McFarling, a portion of a full branch instruction address is used to access a prediction entry in a corresponding structure would be considered hashing.

5. Referring to claim 6, McFarling has taught a processor comprising:

a) an instruction pointer (IP) generator capable of providing an address. See column 3, lines 12-16. Note that McFarling makes reference to a program counter, which performs the same function as an IP generator (i.e. they both provide an address at which an instruction will be fetched).

b) saturating counter branch prediction (SCBP) logic having an input coupled to the IP generator and capable of providing a first taken/not-taken prediction at an output responsive to the address. See column 9, lines 10-16. It has been disclosed that a bimodal predictor always provides a prediction. Furthermore, from column 3, lines 11-28, the bimodal predictor is shown to be an array of saturated counters. Furthermore, column 3, lines 12-16, explain that the SCBP logic is coupled to the program counter (IP generator).

c) local branch history prediction (LBHP) logic having an input coupled to the IP generator and capable of providing (1) a second taken/not-taken prediction at anoutput responsive to the address resulting in a hit, and (2) a hit/miss indication for the address, wherein the LBHP logic includes at least one local branch history table to provide a taken/not-taken history in response to a hit. See column 9, lines 18-24, and Fig.10. Note in Fig.10 that the hit indication controls which prediction is propagated through the multiplexer. Also, in column 4, lines 15-19, McFarling has explained that the local history table is indexed based on a portion of the branch instruction address. Therefore, the local history table is also coupled to the IP generator. It should be realized that the LBHP includes at least one local branch history table to provide a taken/not-taken history in response to a hit. From Fig.10, it can be seen that the tag/history data is stored in one table 100 and the prediction (CNT) is stored in a second table. The most significant bit of CNT is used to make a taken/not-taken prediction.

d) a multiplexer having an input coupled to the outputs of the SCBP and LBHP logic and a select input coupled to receive the hit/miss indication and in response provide (1) the second prediction if there is a hit and (2) the first prediction if there is a miss. See Fig.10 and note that the multiplexer has inputs from both the SCBP (104) and LBHP (104) and that one of the inputs is chosen based on the hit signal that is used as a select line for the multiplexer. See column 9, lines 10-24 for further explanation.

e) entry replacement logic to update a replacement field for a matching entry in the at least one table only if the first prediction is incorrect. See column 9, lines 45-52. When the first prediction from the bimodal predictor (saturating counter predictor) is correct, the corresponding entry in the history table is not updated. Therefore, replacements would only occur when the first prediction is incorrect.

6. Referring to claim 7, McFarling has taught a processor as described in claim 6. McFarling has further taught address hash logic coupled between the IP generator and the inputs of the SCBP and LBHP logic to provide a plurality of index values to at least one of the SCBP and LBHP logic. Again, as discussed in column 3, lines 12-16, and column 4, lines 15-19, the prediction logic is provided an index by performing a truncation hashing function. More specifically, it has been taught (in the above passages) that the prediction logic receives a certain amount of low order branch instruction address bits as an index instead of the entire address. Therefore, the subset of wires used to carry the index from the IP generator to the prediction logic is address hash logic that performs a truncation algorithm. This logic transforms an initial value (branch instruction address) into another value used for locating corresponding data in a structure (SCBP and LBHP).

7. Referring to claim 8, McFarling has taught a processor as described in claim 6. McFarling has further taught that the SCBP logic includes a bimodal predictor. See column 9, lines 10-24.

8. Referring to claim 14, McFarling has taught an apparatus comprising:

a) means for providing an address of at least one instruction. See column 3, lines 12-16. Note that McFarling makes reference to a program counter, which performs the same function as an IP generator (i.e. they both provide an address at which an instruction will be fetched).

b) means for providing a first taken/not-taken branch prediction based upon the current state of a state machine and responsive to the address. See column 9, lines 10-16. It has been disclosed that a bimodal predictor always provides a prediction. Furthermore, from column 3, lines 11-28, the bimodal predictor is shown to be an array of saturated counters. Saturated counters are state machines in that they function differently when in different states. See Fig.2.

c) local branch history prediction (LBHP) logic having an input coupled to the address providing means and capable of providing (1) a second taken/not-taken prediction at an output responsive to the address resulting in a hit, and (2) a hit/miss indication for the address, wherein the LBHP logic includes at least one local branch history table to provide a taken/not-taken history in response to a hit. See column 9, lines 18-24, and Fig.10. Note in Fig.10 that the hit indication controls which prediction is propagated through the multiplexer. Also, in column 4, lines 15-19, McFarling has explained that the local history table is indexed based on a portion of the branch instruction address. Therefore, the local history table is also coupled to the IP generator. It should be realized that the LBHP includes at least one local branch history table to provide a taken/not-taken history in response to a hit. From Fig.10, it can be seen that the tag/history data is stored in one table 100 and the prediction (CNT) is stored in a second table. The most significant bit of CNT is used to make a taken/not-taken prediction.

d) a multiplexer having an input coupled to the outputs of the first prediction means and the LBHP logic and a select input coupled to receive the hit/miss indication and in response provide (1) the second prediction if there is a hit and (2) the first prediction if there is a miss. See Fig.10 and note that the multiplexer has inputs from both the SCBP (104) and LBHP (104) and that one of the inputs is chosen based on the hit signal that is used as a select line for the multiplexer. See column 9, lines 10-24 for further explanation.

e) means for updating a replacement field for a matching entry in the at least one table only if the first prediction is incorrect. See column 9, lines 45-52. When the first prediction from the bimodal predictor (saturating counter predictor) is correct, the corresponding entry in the history table is not updated. Therefore, replacements would only occur when the first prediction is incorrect.

9. Referring to claim 15, McFarling has taught an apparatus as described in claim 14. Furthermore, claim 15 is rejected for the same reasons set forth in the rejection of claim 2. Note that hashing and encoding perform the same operation in that a starting value (branch instruction address) is transformed into another value (index to prediction logic).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over McFarling, as applied above, in view of Hennessy and Patterson, Computer Architecture - A Quantitative Approach, 2nd Edition (herein referred to as Hennessy).

12. Referring to claim 4, McFarling has taught a method as described in claim 1.

a) Furthermore, it is inherent that an instruction for which a prediction is made would have been fetched at the corresponding address. This concept is also supported in column 3, lines 12-16. The bits that are used to index the predictors are taken from an address that is stored in the program counter (PC). The PC is a register that is used to store a pointer to a location in memory from which an instruction is fetched.

b) In addition, McFarling has taught that if the instruction is a branch, a determination as to whether the branch is taken or not-taken will not be available until the instruction has progressed beyond a decode stage. See column 1, lines 33-36, and note that the outcome of the branch is unknown until after the branch is executed. Since execution inherently follows decoding, the branch determination is not known until the instruction has progressed beyond the decode stage.

c) Finally, although it is inherent that McFarling's system decodes fetched instructions (in order to determine what operation would be performed), McFarling has not explicitly taught that at least one of the first and second predictions is available when the at least one instruction is being decoded. However, Hennessy has shown the state of a pipeline that implements branch prediction. See Figure 3.26 on page 167. Note that during the decode stage (ID) of the branch instruction, the predicted instruction (Instruction i+1) is being fetched in the fetch stage (IF). In order for the predicted instruction to be fetched during the decode stage of the branch instruction, the prediction must be available when the branch is being decoded. It can be seen that the pipeline is kept full and free of stalls through the use of branch prediction. Furthermore, from column 1, lines 49-56, McFarling has explained that the purpose of a branch prediction scheme is to keep the pipeline full and free from stalls, which in fact is supported by the teachings of Hennessy. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to make the at least one of the predictions available when the at least one instruction is being decoded. This will ensure that the pipeline will stay full, as is desired by McFarling. It should be further noted that the scheme taught by Hennessy does not only apply to a system in which branch outcomes are known at the end of the decode stage. This can be realized since McFarling has taught a taken/not-taken prediction scheme in which branch outcomes are unknown until after the branch executes (column 1, lines 33-45). The general idea of such a prediction scheme is to keep the pipeline full regardless of when the branch outcome is "officially" known, and if the prediction is available while the branch is being decoded, then the next instruction can be fetched without stalling.

13. Referring to claim 5, McFarling in view of Hennessy has taught a method as described in claim 4. If the at least one instruction is a branch, then:

a) it is inherent that the method further comprises determining a target address of the branch. If a target address were not determined for a conditional branch, then the system would not know where to fetch the next instruction from when the branch is predicted taken.

b) it is inherent that the method further comprises loading an instruction pointer generator with the target address if at least one of the first and second predictions indicates that the branch is to be taken. In order to fetch the target instruction of a predicted-taken branch, the target address must be loaded into the IP generator (also known as the program counter (PC)). The IP generator (PC) is a register that is used to store a pointer to a location in memory from which an instruction is fetched. For the applicant's benefit, please see Hennessy, page 162 and Figure 3.22 on page 163. From the figure it can be seen that the PC (IP generator) is loaded with the output of a multiplexer. Inputs to the multiplexer include the next instruction address and the target address (which is outputted by the ADD logic in the ID stage of the pipeline). If the target address were not stored in the IP generator (PC), then the target instruction could not be fetched from instruction memory (as shown in Figure 3.22 of Hennessy).

14. Referring to claim 11, McFarling has taught a processor as described in claim 6. Furthermore, the processor of claim 11 performs the method described in claim 4. Therefore, claim 11 is rejected for the same reasons set forth in the rejection of claim 4.

15. Referring to claim 12, McFarling has taught a processor as described in claim 11. Furthermore, the processor of claim 12 performs the method described in claim 5. In addition, it should be realized from the rejection of claims 4 and 5 that the decode stage taught by Hennessy has the ability to determine a target address. See Figure 3.22 on page 163. Therefore, claim 12 is rejected for the same reasons set forth in the rejection of claim 5.

16. Referring to claim 18, McFarling has taught an apparatus as described in claim 14. Furthermore, the apparatus of claim 18 performs the method of claim 4. Therefore, claim 18 is rejected for the same reasons set forth in the rejection of claim 4.

17. Referring to claim 19, McFarling has taught an apparatus as described in claim 18. Furthermore, the apparatus of claim 19 performs the method of claim 5. Therefore, claim 19 is rejected for the same reasons set forth in the rejection of claim 5.

18. Claims 9 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over McFarling, as applied above, in view of Gochman et al., U.S. Patent No. 5,842,008 (herein referred to as Gochman).

19. Referring to claim 9, McFarling has taught a processor as described in claim 6.

a) 1) McFarling has taught a single branch history table that provides a tag and a taken/not-taken history associated with the tag in response to a hit. See Fig.10. McFarling has not explicitly taught the concept of using multiple branch history tables to provide a tag and a taken/not taken history associated with the tag in response to a hit. However, Gochman has taught the concept of using multiple memories that are simultaneously accessed based on an index. After each memory produces its respective data, a single value is selected and propagated. See Fig.4a and Fig.5. A person of ordinary skill in the art would have recognized that by using multiple memory tables, as opposed to a single table, the number of index bits used to access an entry in each bank would be reduced, resulting in a reduction in the amount of hardware. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to use multiple branch history tables instead of a single table.

2) In addition, it would have been obvious to one of ordinary skill in the art at the time of the invention to separate the single history table as taught by McFarling into multiple history tables since it has been held that making separable where needed is obvious. See Nerwin v. Erlichman, 168 USPQ 177 (1969). It should be realized that the single-table system serves the same purpose as the multiple-table system in that an index is applied and a corresponding entry is selected. Although multiple entries are selected in response to applying an index to multiple tables (in Applicant's system), the final result is still only a single entry, as taught by McFarling.

b) it is inherent that McFarling's system includes compare logic that is connected to the history table in order to determine the hit/miss indication. A hit indication is given when the branch instruction in question has a corresponding entry in the history table. Therefore, in order to give a hit signal, some logic must be used to compare the entries in the table with the current branch instruction. Since McFarling has not taught multiple history tables, it follows that he has not taught multiple compare logic units that are connected to those tables. However, if multiple tables were used, as discussed in part (a) above, then it would be inherent that multiple compare logic units would be connected to those tables in order to determine if a branch has a corresponding entry.

c) McFarling has not taught a history multiplexer coupled to each of the plurality of tables to provide the history for the hit. However, since McFarling has only taught one table, there is no need to multiplex multiple values from multiple tables. On the other hand, if multiple tables were implemented in McFarling's system, as described in part (a) above, then it would follow that a multiplex system would be needed in order to choose one of the plurality of history values. This concept has been taught by Gochman in Fig.4a (component 320) and Fig.5. When each of the values are retrieved from the memories, the selection logic determines which value will be sent through. Therefore, if multiple tables were used in McFarling's system, in order to ensure only the correct history value is used to make the prediction, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a multiplexer to perform the selection.

d) McFarling has taught combinational logic coupled to an output of the history multiplexer to provide the second taken/not-taken prediction. See Fig.10 (prediction line 102 exits from combination logic that produces the taken/not-taken signal). Since McFarling has not taught multiple history tables and a history multiplexer, it follows that he has not taught combinational logic that is connected to a history multiplexer. However, if multiple tables and a history multiplexer were used, as discussed in parts (a) and (c) above, then it would follow that the combinational logic would be coupled to an output of the history multiplexer to provide the second taken/not-taken prediction. For instance, this concept is also shown in Fig.4a, component 330, of Gochman, which controls the prediction based on the output of the selection logic. Therefore, if multiple tables and a history multiplexer were used, it would have been obvious to one of ordinary skill in the art at the time of the invention to connect the combinational logic taught by McFarling to the output of the history multiplexer since the combinational logic uses the history information in order to generate a taken/not-taken prediction.

20. Referring to claim 16, McFarling has taught an apparatus as described in claim 14. Furthermore, claim 16 is rejected for the same reasons set forth in the rejection of claim 9.

21. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over McFarling, as applied above, in view of Rahman et al., U.S. Patent No. 5,805,878 (herein referred to as Rahman).

22. Referring to claim 13, McFarling has taught a processor as described in claim 6. McFarling has not explicitly taught that the address points to a cache line having a plurality of instructions. However, Rahman has taught the implementation of a cache with multiple instructions per cache line. See column 9, lines 17-21. Also, Rahman has suggested in column 9, lines 23-37, that a benefit of such a feature would be to allow for simultaneous fetching of multiple instructions. This would result in less time spent making memory accesses and increased instruction-level parallelism, which would result in higher throughput. Therefore, in order to increase throughput, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a cache that contains a plurality of instructions per cache line.

Response to Arguments

23. Applicant's arguments filed on April 29, 2003, have been fully considered but they are not persuasive.

24. In the remarks, Applicant argues the rejection of claim 1 on pages 9-10, in substance that:

> McFarling "does not teach or suggest updating a replacement field for a matching entry in a local branch history table only if a prediction using a saturating counter branch predictor is incorrect, indicating that the entry is used to make a prediction."

25. These arguments are not found persuasive for the following reasons:

a) In amended claim 1, it is claimed that a replacement field is updated only if the first (global) prediction is incorrect. McFarling, in column 9, lines 45-55, has disclosed that a replacement field is not updated if the first (global) prediction is correct. These two can be considered equivalents in that with McFarling, if the replacement field is not updated when the first prediction is correct, then the replacement field is updated only when the first prediction is incorrect (as claimed by applicant).

26.. In the remarks, Applicant argues the rejection of claim 4 on page 11, in substance that:

> "...in this particular pipeline (Hennessy), the branch is actually taken or not taken at the instruction decode stage. In contrast, applicant has clarified that claim 4 is directed to the situation where if the instruction is a branch, a determination as to whether the branch is taken or not taken will not be available until the instruction has progressed beyond the decode stage. Accordingly, the situation described in Hennessy and cited by the examiner is a clearly different scenario where the branch is either taken or not taken at the decode stage rather that in a later stage of the pipeline.
>
> In addition, the particular scenario described in Hennessy is not similar to the techniques described in McFarling because in Hennessy the prediction scheme is explicitly assumed to predict every branch as not taken, allowing the hardware to continue as if the branch were not executed. Accordingly, applicant submits that Hennessy does not fairly teach or suggest to one of ordinary skill in the art faced with the mechanisms of McFarling that

> the timing of the pipeline stages be modified to support the method recited in applicant's claim 4."

27. These arguments are not found persuasive for the following reasons:

a) This amendment to the claim 4 is anticipated by McFarling in that McFarling has disclosed, in column 1, lines 33-36, that the outcome of a branch instruction is unknown until after the branch instruction executes. Since execution inherently follows decoding, the branch outcome is unknown until the branch progresses beyond a decode stage (in addition, applicant should note that Hennessy has also taught that branch instruction outcomes may not be known until they are beyond the decode stage. See Fig.3.21. It should be further noted that the prediction scheme taught by Hennessy does not only apply to a system in which branch outcomes are known at the end of the decode stage. This can be realized since McFarling has taught a taken/not-taken prediction scheme in which branch outcomes are unknown until after the branch executes (column 1, lines 33-45). The general idea of such a prediction scheme is to keep the pipeline full regardless of when the branch outcome is "officially" known, and if the prediction is available while the branch is being decoded, then the next instruction can be fetched without stalling, thereby maximizing the work done in the pipeline.

b) Furthermore, it should be realized that the Hennessy reference is used to merely show that a branch prediction (in this specific instance, a not-taken prediction) can be provided while the branch instruction is in the decode stage so that instructions can be fetched and the pipeline can be kept full. A person of ordinary skill in the art would have recognized that a taken prediction could also be provided at the same stage. Therefore, the Hennessy reference should not be viewed as being different from McFarling because it only shows a not-taken prediction. Instead, it should be viewed as a reference used to show the advantages of a prediction being made while a branch instruction is in the decode stage. It is in this context that the Hennessy reference applies to applicant's claims.

Conclusion

28. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

DJH
David J. Huisman
June 9, 2003

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100